

ipd1100madbvrnTES-10

**Defense Information Infrastructure (DII)
Common Operating Environment (COE)**

**Application Program Interface Reference Manual (APIRM)
for the
Digitized Bathymetric Database API (MADBV) Segment
of the
Tactical Environmental Support System Next Century
[TESS(NC)]
Meteorology and Oceanography (METOC) Database**

Preliminary Release

Document Version 1.1

2 October 1998

**Prepared for:
Naval Research Laboratory
Marine Meteorology Division
Monterey, CA**

**Prepared by:
Integrated Performance Decisions, Inc.
Middletown, RI**

PRINTED COPY IS UNCONTROLLED AND MAY BE OBSOLETE

ipd1100madbvrnTES-10

Table of Contents

1	SCOPE.....	1
1.1	Identification	1
1.2	System Overview	1
2	REFERENCED DOCUMENTS	5
2.1	Government Documents	5
2.2	Non-Government Documents.....	6
3	MADBV OVERVIEW.....	7
3.1	Overview of the Digitized Bathymetric Data (MDDBV) Segment	7
3.2	API Overview	8
3.3	Bathymetry Data Structures/Defines	9
3.3.1	Bathymetry Defines/Enumerations	9
3.3.2	Bathymetry Area of Interest (AOI) Structure.....	10
3.3.3	Track Request Structure.....	10
3.3.4	Track Data Structure	10
3.3.5	Contour Description Structure	11
3.3.6	Contour Segment Structure.....	11
3.3.7	Bathymetry Return Structure	11
3.4	MADBV Error Definitions.....	12
4	CONNECT APIs.....	13
4.1	MADBVConnect	14
4.2	MADBVDisconnect.....	15
5	RETRIEVAL APIs.....	17
5.1	MADBVGetCatalog.....	17
5.2	MADBVGetTrack.....	19
5.3	MADBVGetGrid	21
5.4	MADBVGetContours	23
6	MADBV UTILITY METHODS AND FUNCTIONS	25
6.1	MADBVFreeContours.....	25
7	NOTES	27
7.1	Glossary of Acronyms.....	27

List of Figures

1-1	TESS(NC) METOC Database - DII COE Segment View	3
1-2	Distributed APIs via COTS RDBMS Client/Server Functionality	3
1-3	Distributed APIs via DII COE Kernel Services (NFS)	4

1 SCOPE

1.1 Identification

This Application Program Interface (API) Reference Manual (APIRM) describes the APIs provided in the Digitized Bathymetry (Variable Resolution) API (MADBV) segment, Version 1.1.0.0, of the TESS(NC) Meteorology and Oceanography (METOC) Database. The MADBV segment provides APIs for the retrieval of historical bathymetry data. This software is designed to run under the Defense Information Infrastructure (DII) Common Operating Environment (COE), release 3.1, on a Hewlett-Packard computer running HP-UX 10.20 or a personal computer running the Microsoft Windows NT 4.0 operating system with Service Pack 3.

1.2 System Overview

The APIs described in this document form a portion of the METOC Database component of the TESS(NC) Program (Navy Integrated Tactical Environmental Subsystem (NITES) Version I). On 29 October 1996, the Oceanographer of the Navy issued a TESS Program Policy statement in letter 3140 Serial 961/6U570953, modifying the Program by calling for five seamless software versions that are DII COE compliant, preferably to level 5.

The five versions are:

- NITES Version I The local data fusion center and principal METOC analysis and forecast system (TESS(NC))
- NITES Version II The subsystem on the Joint Maritime Command Information System (JMCIS) or Global Command and Control System (GCCS) (NITES/Joint METOC Segment (JMS))
- NITES Version III The unclassified aviation forecast, briefing, and display subsystem tailored to Naval METOC shore activities (currently satisfied by the Meteorological Integrated Data Display System (MIDDS))
- NITES Version IV The Portable subsystem composed of independent PCs/workstations and modules for forecaster, satellite, communications, and Integrated Command, Control, Communications, Computer, and Intelligence Surveillance Reconnaissance (IC4ISR) functions (currently the Interim Mobile Oceanographic Support System (IMOSS))
- NITES Version V Foreign Military Sales (currently satisfied by the Allied Environmental Support System (AESS))

NITES I acquires and assimilates various METOC data for use by US Navy and Marine Corps weather forecasters and tactical planners. NITES I provides these users with METOC data, products, and applications necessary to support the warfighter in tactical operations and decision making. NITES I provides METOC data and products to NITES I and II applications, as well as non-TESS(NC) systems requiring METOC data, in a heterogeneous, networked computing environment.

The TESS(NC) Concept of Operations and system architecture require that the METOC Database be distributed both in terms of application access to METOC data and products and in terms of physical location of the data repositories. The organizational structure of the database is influenced by these requirements, and the components of this distributed database are described below.

In accordance with DII COE database concepts, the METOC Database is currently composed of five DII COE-compliant *shared database* segments and one DII COE-compliant data segment. Associated with each shared database and data segment is an API segment. This organization is shown in Figure 1-1. The segments are arranged by data type as follows:

<u>Data Type</u>	<u>Data Segment</u>	<u>API Segment</u>
Grid Fields	MDGRID	MAGRID
Latitude-Longitude-Time (LLT) Observations	MDLLT	MALLT
Textual Observations and Bulletins	MDTXT	MATXT
Remotely Sensed Data	MDREM	MAREM
Imagery	MDIMG	MAIMG
Historic Bathymetry Data	MDDBV	MADBV

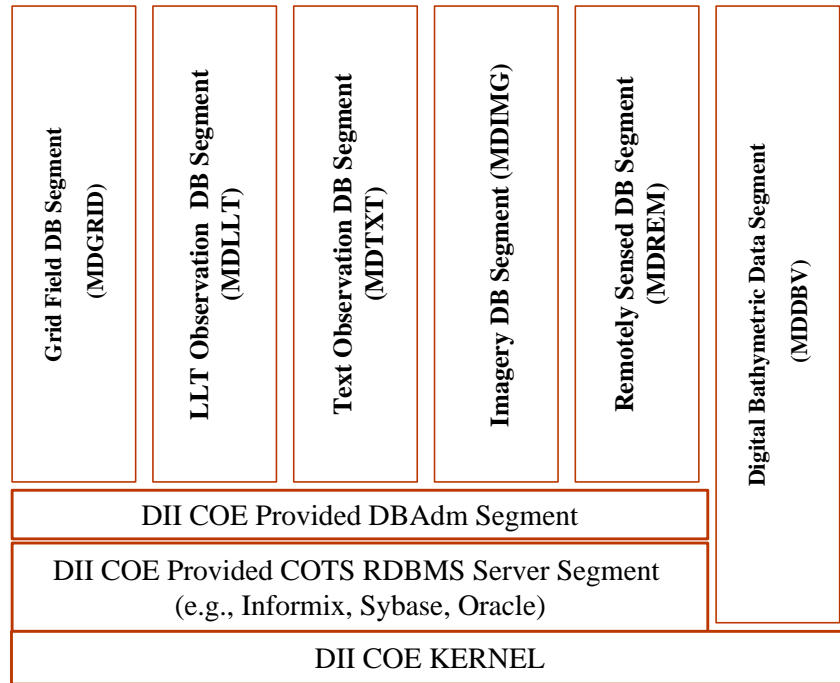


Figure 1-1. TESS(NC) METOC Database - DII COE Segment View

Typical client-server installations access shared database segments via a COTS RDBMS client/server as shown in Figure 1-2. This shows the shared database segments residing on a DII COE SHADE database server, with a NITES I or II client machine hosting the API segments. Communication between API segments and shared database segments is accomplished over the network using ANSI-standard Structured Query Language (SQL).

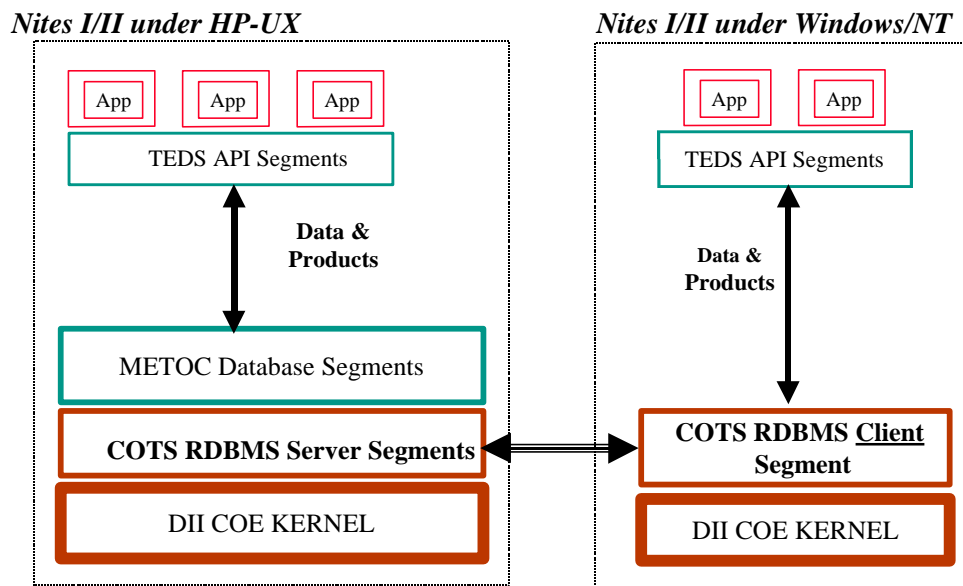


Figure 1-2. Distributed APIs via COTS RDBMS Client/Server Functionality

Data Segments are static files of historic data. DII COE data segments are available over a distributed network via DII COE Kernel Service (NFS). In this case, the data segments are accessed directly by the distributed APIs (Figure 1-3). The platform running the applications needing the data must first mount the file system containing the data segment. The remote system may then access the data from the mounted drive using NFS services. Access to the mounted drive is then transparent to the application/API utilizing the data.

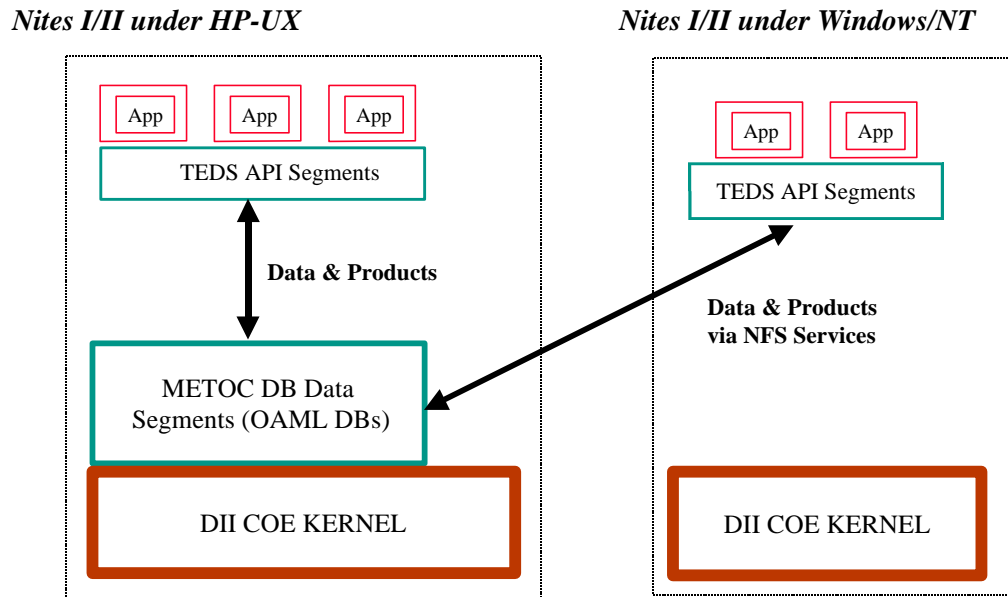


Figure 1-3. Distributed APIs via DII COE Kernel Services (NFS)

The MADBV segment deals with historic bathymetry data. The data are generated from the Oceanographic and Atmospheric Master Library (OAML) Digitized Bathymetric Data Base – Variable resolution (DBDB-V) data and provide global water depth at various resolutions throughout the world.

ipd1100madbvpmTES-10 2 October 1998	<i>Programming Manual (PM) for the Digitized Bathymetry API (MADBV) Segment of the Tactical Environmental Support System Next Century [TESS(NC)] Meteorology and Oceanography (METOC) Database</i>
DII.COE31.HP10.20.CIP 23 May 1997	<i>DII COE V3.1 HP 10.20 Consolidated Installation Procedures</i>
DII.3010.HP1020.KernelP1.IG-1 9 May 1997	<i>DII COE Kernel 3.0.1.0P1 Patch 1 for HP-UX 10.20 Installation Guide</i>
DII.3010.HP1020.KernelP2.IG-1 30 July 1997	<i>DII COE Kernel 3.0.1.0P2 Patch 2 for HP-UX 10.20 Installation Guide</i>
DII.3010.HP1020.KernelP3.IG-1 08 August 1997	<i>DII COE Kernel 3.0.1.0P3 Patch 3 for HP-UX 10.20 Installation Guide</i>
DII.3010.HP1020.KernelP4.IG-1 27 August 1997	<i>DII COE Kernel 3.0.1.0P4 Patch 4 for HP-UX 10.20 Installation Guide</i>
Unnumbered February 1997	<i>Database Description for Digital Bathymetric Database – Variable Resolution (DBDB-V) Version 1.0</i>

2.2 Non-Government Documents

World Meteorological Organization, Geneva, Switzerland

WMO-306	<i>Manual On Codes</i>
---------	------------------------

3 MADBV OVERVIEW

3.1 Overview of the Digitized Bathymetric Data (MDDBV) Segment

The Digitized Bathymetric Data segment consists of a static data set developed by NAVOCEANO, and a pair of bathymetry contour files generated from the NAVOCEANO data set. The Ocean Floor Depth Digital Bathymetric Data Base Variable Resolution (DBDB-V) is a digital bathymetric database that provides ocean depths at various gridded resolutions.

The NAVOCEANO developed data set for DBDB-V consists of four file types. The depth information is expressed in meters, uncorrected at an assumed sound velocity of 1500 m/s. The file types are:

- **Master Index File** – Contains a pointer or byte address to each populated 1 deg cell for each of the available resolutions.
- **Index File** – Provides a link to the detailed depth values, as well as a link to a description file associated with the depths.
- **Description File** – Provides details on the compressions, scaling, and storage of the depth information.
- **Data File** – Contains the depth values for a 1-deg cell of a specific resolution.

Two additional data files have been generated using the DBDB-V data in order to provide bathymetry contours suitable for display on a chart product. The contour data is a static global bathymetry database consisting of non-overlapping tiles of 2 degrees on a side. Tiles for areas South of 78 S were not generated, so the indexing starts at 78 S, 0 E. The file types are:

- **Index File** – Contains the number of contour levels available, and for each tile, the details each contour level and the address of the contour data in the contour data file.
- **Contour Data File** – Contains the contour segments for each contour level.

This DII COE compliant Data Segment must be installed on a file system that can be mounted by the platform running the applications that require this data. Unlike DII COE database segments, data segments are accessed directly by the associated APIs. This requires that the Data Segment be visible to the application accessing it. This visibility is accomplished by means of an NFS mount of the volume containing the data segment (Figure 1-3). If an application utilizing the MDDBV segment resides on a different machine than the database, access to the database is available via NFS services if the drive is mounted.

3.2 API Overview

There are seven public APIs utilized to access and manage the bathymetric data segment.

1. **MADBVConnect:** Establish connection to the bathymetry database.
2. **MADBVDisconnect:** Terminate the connection with the bathymetry database.
3. **MADBVGetCatalog:** Retrieve a list of the data resolutions available within the bathymetry data segment.
4. **MADBVGetTrack:** Retrieve the bathymetry along a line of bearing (track).
5. **MADBVGetGrid:** Retrieve bathymetry for a gridded area.
6. **MADBVGetContours:** Retrieve available bathymetry contours for a given area.
7. **MADBVFreeContours:** Free the memory allocated for the contours.

These APIs require that the MDDBV Data Segment has been installed on, or mounted by the platform on which applications using these APIs are running. The MDDBV data segment is configured as global data. The DATA_DIR environment variable must be known on the target system.

3.3 Bathymetry Data Structures/Defines

The following structures are used by the Grid Field APIs and are provided for reference:

3.3.1 Bathymetry Defines/Enumerations

The following defines are found in MADBVDefines.h They provide a reference for most of the conversions and geographic calculations performed within the extraction APIs.

```
#define MAX_RESOLUTION      20
#define NO_DATA             -1
#define CHRTRNULL          10000000000000000.0
#define DB_NULL            32767
#define MAX_TRACK           1000
#define METERS_TO_FT        .3048
#define INDEX_REC_LENGTH   37
#define NUM_CELLS           65160
#define DB_NONE             -999
#define TRUE                1
#define FALSE               0
#define LAT_MILES           60.0
#define EARTH_RAD           3440.09
#define RAD_TO_DEG          57.29577951
#define DEG_TO_RAD           .01745329252
```

The following defines are found in MADBVAPI.h. The define for LAND is the value returned if a requested bathymetry point is on land.

```
#define FILELENGTH          132
#define LAND                 -10
```

The following typedefs, found in MADBVAPI.h, enumerate specific types used in the requests to the APIs, and are meant to make the code more readable and the interface more intuitive.

```
typedef long                MADBVCONTEXT, *PMADBVCONTEXT;

typedef enum
{
    Minutes,
    NauticalMiles,
    DatabaseBest
} ExtractType;

typedef enum
{
    Meters,
    Feet,
    Fathoms
} DataUnits;
```

3.3.2 Bathymetry Area of Interest (AOI) Structure

This structure defines the Geographical Area of interest. This structure is defined in MADBVAPI.h. This structure is used to define the area to be extracted for gridded bathymetry and bathymetric contour lookups.

```
typedef struct gatmadbvGeoArea
{
    float  rsNLat;      /* North Latitude */
    float  rsSLat;      /* South Latitude */
    float  rsWLon;      /* West Longitude */
    float  rsELon;      /* East Longitude */
} MADBVGEOAREA, *PMADBVGEOAREA;
```

3.3.3 Track Request Structure

This structure defines the request for bathymetry along a track (Line of Bearing). This structure is defined in MADBVAPI.h.

```
typedef struct tagmadbvTrackQuery
{
    float      rsRange;      /* range of track in nautical miles */
    float      rsBearing;    /* bearing of track in degrees */
    float      rsLat;        /* starting latitude */
    float      rsLon;        /* starting longitude */
    ExtractType nExtractionType; /* extraction type needed */
    DataUnits  nExtractionUnits; /* units for return data */
    int        nLandCheckFlag; /* 0 -> Terminate extraction when */
                                /* land is encountered. */
                                /* 1 -> return data to end of track */
    float      rsResolution; /* extraction resolution */
                                /* - not used if ExtractionType is */
                                /* DatabaseBest */
} MADBVTRACKQUERY, *PMADBVTRACKQUERY;
```

3.3.4 Track Data Structure

The Track Data structure contains the Latitude, Longitude, range from the beginning of the track (nm), and the bathymetry for each data point retrieved along the track. The lat/lon and range information is required due to the fact that the retrieved data is not necessarily equally spaced in range or decimal degrees for the length of the retrieval. The Track request returns an array of structures with the following format. This structure is defined in MADBVAPI.h

```
typedef struct tagmadbvTrackData
{
    float  rsRange;      /* range in nautical miles */
    float  rsBathymetry; /* Water depth in requested units */
    float  rsGCLatitude; /* great circle latitude */
    float  rsGCLongitude; /* great circle longitude */
} MADBVTRACKDATA, *PMADBVTRACKDATA;
```

3.3.5 Contour Description Structure

The contour description structure describes the contours retrieved for a given area. For each contour level, depth contains the contour depth in fathoms, numSeg gives the number of contour segments, and pContour points to an array of numSeg contour segment structures. This structure is defined in MADBVAPI.h

```
struct contour_desc
{
    int          depth;
    int          numSeg;
    PCNTR_LINE_STRUCT  pContour;
};
typedef struct contour_desc CONTOURDESC, *PCONTOURDESC;
```

3.3.6 Contour Segment Structure

The contour segment structure gives the start/end latitude/longitude of a portion of a bathymetry contour. This structure is defined in MADBVAPI.h

```
struct cntr_line_struct
{
    float lat[2];
    float lon[2];
};
typedef struct cntr_line_struct CNTR_LINE_STRUCT, *PCNTR_LINE_STRUCT;
```

3.3.7 Bathymetry Return Structure

Each of the Bathymetry Data APIs returns this structure containing status data. The *nStatus* field will contain a zero upon successful completion of the API call. Any value of nStatus greater than one indicates an MADBV error that maps to a define in the file MADBVErr.h. The Bathymetry return structure is defined in MADBVAPI.h.

```
typedef struct tagmadbvRet
{
    int          nStatus;          /* if nStatus == 0 >> Success */
                                   /* if nStatus == 2 >> Error */
    char          szSQLState[6];    /* Not used */
    char          szErrorMessage[290]; /* String containing information */
                                   /* about the error */
} MADBVRET, *PMADBVRET;
```

3.4 MADBV Error Definitions

The following are the MADBV error definitions contained in the file MADBVErr.h.

```
#ifndef MADBVERR_OFFSET
#define MADBVERR_OFFSET    10000
#endif

#define MADBV_MEMORY_ERR      1 + MADBVERR_OFFSET
#define MADBV_INDEX_FILE_ERR  2 + MADBVERR_OFFSET
#define MADBV_MASTER_FILE_ERR 3 + MADBVERR_OFFSET
#define MADBV_DESCRIP_FILE_ERR 4 + MADBVERR_OFFSET
#define MADBV_DATA_ERR        5 + MADBVERR_OFFSET
#define MADBV_HOME_ERR        6 + MADBVERR_OFFSET
#define MADBV_CONTEXT_ERR     7 + MADBVERR_OFFSET
#define MADBV_LATLON_ERR      8 + MADBVERR_OFFSET
#define MADBV_AREA_ERR        9 + MADBVERR_OFFSET
#define MADBV_SIZE_ERR       10 + MADBVERR_OFFSET
#define MADBV_CON_INDEX_ERR  11 + MADBVERR_OFFSET
#define MADBV_CON_DATA_ERR   12 + MADBVERR_OFFSET
```

4 CONNECT APIs

The connect APIs are used to establish or disestablish a connection with the data within the MDDBV Data Segment. This connection requires that the file system containing the MDDBV segment be mounted on the local platform, and the DATA_DIR environment variable be known.

Information about each API is presented in manual page format as follows:

NAME

Function Name – Provides a brief description of the function.

SYNOPSIS

Presents the calling syntax for the routine, including the declarations of the arguments and the return type. Also lists the necessary include files for each routine.

INPUT PARAMETERS

Describes each of the input parameters used by the function.

OUTPUT PARAMETERS

Describes each of the parameters output by the function.

DESCRIPTION

Describes what the function does and what events or side effects it causes.

RETURNS

Describes what the function returns.

NOTE

Provides any applicable notes about the function.

SEE ALSO

Provides a reference to related functions.

Examples showing the proper usage of the APIs are presented in the *Digitized Bathymetry API Programming Manual*, referenced in Section 2.

4.1 MADBVConnect

NAME

MADBVConnect – Connects the application to the database.

SYNOPSIS

```
#include "MADBVAPI.h"
#include "MADBVErr.h"
MADBVRET MADBVConnect(PMADBVCONTEXT *pContext,
                      long lAccessFlag);
```

INPUT PARAMETERS

lAccessFlag – flag specifying file access type for retrievals.
0 if files are to be opened and closed each retrieval.

*pContext – Address of a pointer to type PMADBVCONTEXT

OUTPUT PARAMETERS

pContext – Database context for bathymetry retrievals.

DESCRIPTION

This subroutine connects the application to the database.

RETURNS

MADBVRET structure – See Section 3.3.7 for details.

NOTE

1. This interface must be called before any other MADBV APIs may be called. It should only be called once per session.

SEE ALSO

MADBVDisconnect (Section 4.2)

4.2 MADBVDisconnect

NAME

MADBVDisconnect – Disconnects the application from the database.

SYNOPSIS

```
#include "MADBVAPI.h"  
#include "MADBVErr.h"  
MADBVRET MADBVDisconnect (PMADBVCONTEXT *pContext);
```

INPUT PARAMETERS

pContext – address of the database context.

OUTPUT PARAMETERS

None.

DESCRIPTION

This subroutine disconnects the application from the database, ending the database session.

RETURNS

MADBVRET structure – See Section 3.3.7 for details.

NOTE

1. MADBVDisconnect should be called to end each database session started by MADBVConnect. It should be called once per session.

SEE ALSO

MADBVConnect (Section 4.1)

(This page intentionally left blank.)

5 RETRIEVAL APIs

This section describes the APIs that are used to retrieve data from the MDDBV database.

5.1 MADBVGetCatalog

NAME

MADBVGetCatalog – Retrieves an array of data resolutions available within the database.

SYNOPSIS

```
#include "MADBVAPI.h"
```

```
#include "MADBVErr.h"
```

```
MADBVRET MADBVGetCatalog( PMADBVCONTEXT pContext,  
                           float **prCatalog,  
                           int *nNumResolutions)
```

INPUT PARAMETERS

PMADBVCONTEXT pContext – A pointer to the Data context established at connection.

float **prCatalog – The address at which to store a pointer to an array of available resolutions.

int *nNumResolutions – The address at which to store the number of resolutions available.

OUTPUT PARAMETERS

int nNumResolutions – Number of resolutions available in the database.

float *prCatalog – Array containing the resolutions (in decimal minutes) available within the database.

DESCRIPTION

This subroutine determines how many resolutions are in the database, and what the resolutions are. The address of memory allocated to hold the resolution data is returned to the user in prCatalog, and the number of resolutions available is stored in nNumResolutions.

RETURNS

MADBVRET structure – See Section 3.3.7 for details.

NOTES

1. MADBVConnect must have been called to start a database session before MADBVGetCatalog may be called.
2. The calling program is responsible for freeing prCatalog when the data is no longer needed.

SEE ALSO

MADBVConnect (Section 4.1)

5.2 MADBVGetTrack

NAME

MADBVGetTrack – Retrieves bathymetry along the specified line of bearing.

SYNOPSIS

```
#include "MADBVAPI.h"
#include "MADBVErr.h"
MADBVRET MADBVGetTrack(PMADBVCONTEXT pContext,
                        PMADBVTRACKQUERY pQuery,
                        int *nNumPoints,
                        PMADBVTRACKDATA *pTrackData);
```

INPUT PARAMETERS

- | | |
|-----------------------------|--|
| PMADBVCONTEXT pContext | – Pointer to the database context established at the connection. |
| PMADBVTRACKQUERY pQuery | – A pointer to a MADBVTRACKQUERY structure (Section 3.3.3) |
| int *nNumPoints | – The address where the number of retrieved points is to be stored. |
| PMADBVTRACKDATA *pTrackData | – The address of a pointer to a MADBVTRACKDATA structure (Section 3.3.4) where the pointer to the retrieved data will be stored. |

OUTPUT PARAMETERS

- | | |
|----------------------------|---|
| int nNumPoints | – The number of bathymetry points retrieved. |
| PMADBVTRACKDATA pTrackData | – A pointer to a MADBVTRACKDATA structure (Section 3.3.4), containing the first of nNumPoints of retrieved bathymetry data. |

DESCRIPTION

MADBVGetTrack retrieves the bathymetry along a track using the parameters specified within the MADBVTRACKQUERY structure (Section 3.3.3). If a valid connection exists, the data is extracted from the bathymetry database at the desired resolution/spacing, and returned to the user as an array of PMADBVTRACKDATA structures.

RETURNS

MADBVRET structure (Section 3.3.7)

NOTES

1. MADBVConnect must have been called to start a database session before MADBVGetTrack may be called.
2. The calling program is responsible for freeing pTrackData when the data is no longer needed.

SEE ALSO

MADBVConnect (Section 4.1)

5.3 MADBVGetGrid

NAME

MADBVGetGrid - Retrieves a grid of bathymetry points.

SYNOPSIS

```
#include "MADBVAPI.h"
#include "MADBVErr.h"
MADBVRET MADBVGetGrid (PMADBVCONTEXT pContext
                        PMADBVGEOAREA pArea,
                        DataUnits nExtractionUnits,
                        float rsResolution,
                        int *nNumRows,
                        int *nNumCols,
                        float **prsBathymetry)
```

INPUT PARAMETERS

PMADBVCONTEXT pContext	– A pointer to the database context established at connection.
PMADBVGEOAREA pArea	– A pointer to a MADBVGEOAREA structure.
DataUnits nExtractionUnits	– Desired units for the extracted data.
float rsResolution	– Desired resolution at which to extract the data.
int *nNumRows	– Address of the location to store the number of extracted rows of bathymetry.
int *nNumCols	– Address of the location to store the number of extracted columns of bathymetry.
float **prsBathymetry	– Address at which to store the pointer to the retrieved bathymetry grid.

OUTPUT PARAMETERS

int nNumRows	– Number of rows of bathymetry retrieved.
int nNumCols	– Number of columns of bathymetry retrieved.
float *prsBathymetry	– Address where the bathymetry grid is stored.

DESCRIPTION

MADBVGetGrid provides bathymetry to the calling program at the resolution specified in rsResolution. The area for which bathymetry is to be extracted is specified in the MADBVGEOAREA structure (Section 3.3.2), and the data is converted to the units specified in nExtractionUnits (see Section 3.3.1). The function outputs the number of rows and columns worth of data retrieved (row/column ordering), and a pointer to the retrieved data. Point (0,0) of the retrieved data is for the lower-left hand corner of the requested area.

RETURNS

MADBVRET structure (Section 3.3.7)

NOTES

1. MADBVConnect must have been called to establish the connection to the database and to generate a valid database context before MADBVGetGrid can be called.
2. The calling program is responsible for freeing the memory referenced by prsBathymetry when the data is no longer needed.

SEE ALSO

MADBVConnect (Section 4.1)

5.4 MADBVGetContours

NAME

MADBVGetContours – Retrieves all available bathymetric contours for a specified area.

SYNOPSIS

```
#include "MADBVAPI.h"
#include "MADBVErr.h"
MADBVRET MADBVGetContours(PMADBVGEOAREA pArea,
                           int *nNumContours,
                           PMADBVCONTOURDESC *pContourDesc)
```

INPUT PARAMETERS

- | | |
|--|---|
| PMADBVGEOAREA pArea | – Pointer to a MADBVGEOAREA (3.3.2) structure containing the limits of the extraction area. |
| int *nNumContours | – Address where the number of contours retrieved is to be stored. |
| PMADBVCONTOURDESC *pContourDesc | – Address at which the pointer to the array of contour data description structures is to be stored. |

OUTPUT PARAMETERS

- | | |
|----------------------------------|--|
| int nNumContours | – Number of contours retrieved. |
| PCONTOURDESC pContourDesc | – Pointer to the first of nNumContours elements in an array of CONTOURDESC structures. |

DESCRIPTION

Given the area specified by the **MADBVGEOAREA** structure (Section 3.3.2), **MADBVGetContours** reads the contoured bathymetry database and extracts all the contours available within the give area. It provides the calling routine with the number of contour levels available, and an array of **MADBVCONTOURDESC** structures (Section 3.3.5) **nNumContours** long. The contour description structure contains the contour depth level (fathoms), the number of segments comprising that contour, and a pointer to an array of contour segment structures (Section 3.3.6) number of segments long.

RETURNS

MADVRET structure (Section 3.3.7)

NOTES

1. The returned contour description data must be freed with a call to MADBVFreeContours

SEE ALSO

MADBVFreeContours (Section 6.1)

6 MADBV UTILITY METHODS AND FUNCTIONS

6.1 MADBVFreeContours

NAME

MADBVFreeContours – Frees the contour data associated with MADBVGetContours

SYNOPSIS

```
#include "MADBVAPI.h"
#include "MADBVErr.h"
MADBVRET MADBVFreeContours(int nNumContours,
                           PMADBVCONTOURDESC pContourDesc);
```

INPUT PARAMETERS

int nNumContours – Number of contours as returned by MADBVGetContours.

PMADBVCONTOURDESC pContourDesc – The pointer to the array of MADBVCONTOURDESC structures created by MADBVGetContours.

OUTPUT PARAMETERS

None.

DESCRIPTION

MADBVFreeContours will free all the allocated contour segments for each contour level, as well as freeing the memory for the MADBVCONTOURDESC structures.

RETURNS

MADBVRET structure (Section 3.3.7)

SEE ALSO

MADBVGetContours (Section 5.4)

(This page intentionally left blank.)

7 NOTES

7.1 Glossary of Acronyms

AESS	Allied Environmental Support System
AOI	Area of Interest
API	Application Program Interface
APIRM	API Reference Manual
COE	Common Operating Environment
DBDB-V	Digitized Bathymetric Data Base – Variable resolution
DII	Defense Information Infrastructure
GCCS	Global Command and Control System
IC4ISR	Integrated Command, Control, Communications, Computer, and Intelligence Surveillance Reconnaissance
IMOSS	Interim Mobil Oceanographic Support System
JMCIS	Joint Maritime Command Information System
JMS	Joint METOC Segment
LLT	Latitude-Longitude-Time
MADBV	METOC Digitized Bathymetry API Segment

MAGRID	METOC Grid Field API Segment
MATXT	Textual Observations API Segment of the TESS(3)/NC METOC Database
MDDBV	METOC Digitized Bathymetry Data Segment
MDGRID	METOC Grid Field Database Segment
MDTXT	Textual Observations Data Segment of the TESS(3)/NC METOC Database
METOC	Meteorology and Oceanography
MIDDS	Meteorological Integrated Data Display System
OAML	Oceanographic and Atmospheric Master Library
PM	Programming Manual
PS	Performance Specification
SQL	Structured Query Language
TEDS	Tactical Environmental Data System
TESS	Tactical Environmental Support System
TESS(NC)	Tactical Environmental Support System Next Century